TA TRIUMPH-ADLER

TA 1600/30
TA 1600/20
TA 1610/10
COBOL REFERENZKARTEN

Nov. 82

Inhaltsverzeichnis

Vollständiges Sprachgerüst des TA 1600 - COBOL

Allgemeines Format der IDENTIFICATION DIVISION 1

Allgemeines Format der ENVIRONMENT DIVISION 2

Allgemeines Format der Dateisteuererklärung 3

Allgemeines Format der DATA DIVISION 5

Bildschirm-Dateierklärung 6

Allgemeines Format der Datenbeschreibung 7

Allgemeines Format der PROCEDURE DIVISION 9

Allgemeines Format der Verben 10

Allgemeines Format der Bedingungen u. d. COPY-Anweisg. 18

Reservierte Wörter 21

Zusätzlich im ANSI reservierte Wörter 23

Vorrangregeln für PICTURE-Zeichen 24

Klassen und Kategorien der Datenfelder 25

Erlaubte Operanden der MOVE-Anweisung 26

Kombination der Operanden in der SET-Anweisung 26

Dateiorganisationen

Bedeutung der Zustandsschlüssel für sequentielle Dateien 27

Tabelle der zulässigen Anweisungen (sequentiell) 27

Bedeutung der Zustandsschlüssel für indexsequentielle Dateien 28

Tabelle der zulässigen Anweisungen (indexsequentiell) 29

Bedeutung der Zustandsschlüssel für relative Dateien 30

Tabelle der zulässigen Anweisungen (relativ) 31

Bedeutung der Zustandsschlüssel für SCREEN-Verarbeitung 32

Tabelle der erlaubten I/O-Befehle (Bildschirm-Modus) 32

Inhaltsverzeichnis

Bedienung des TA 1630 COBOL-Systems

Ausführung des COBOL-Compilers im Vordergrund - XCCF 33

Flußdiagramm 34

Wahlfreie Angaben für den COBOL-Compiler 35

Ausführung des COBOL-Compilers im Hintergrund - XCC 35

Ausführung eines COBOL-Programmes im Vordergrund - XCPF 35

Ausführung eines COBOL-Programmes im Hintergrund - XCP 35

Erlaubte Link-Editor-Anweisungen bei COBOL-Objektprogrammen 36

COBOL-Laufzeitsystem 37

XCTF-Kommando 37

XCT-Kommando 37

Ausführungscodes von COBOL-Programmen 37

Bedienung des TA 1620 COBOL-Systems

Compilierung an der TA 1620 (TA 1620 COBOL-Compiler) 39

Compilierung an der TA 1630 (Crosscompiler) 39

Programmcompilierung 39

Transfer des Objektcodes 40

Compilierung und Transfer mit dem Kommando XCCRCT (F) 40

Ausführung von COBOL-Programmen an der TA 1620 40

COBOL-Laufzeit-Debug

Debug-Kommandos 41

Fehlermeldungen des Debug-Systems 42

Fehlermeldungen des COBOL-Compilers 43

TA 1620 COBOL-Systemfehlermeldungen 49

Fehlermeldungen des TA 1600 COBOL Laufzeitsystems 53

TA 1630 COBOL-Laufzeit Systemfehlermeldungen 58

Merkmale des TA1600-COBOL

Mit TA1600 COBOL wird in diesem Handbuch die auf der TA1600-Systemfamilie implementierte COBOL Basissprachmenge bezeichnet. Die auf den Systemen TA1610, TA1620 und TA1630 implementierten TA1610 COBOL, TA1620 COBOL und TA1630 COBOL umfassen diese Basissprachmenge.

TA1620/TA1630

TA1620 bzw. TA1630 COBOL enthält zusätzliche Erweiterungen, die nicht in TA1610 COBOL verfügbar sind; diese sind durch Umrahmungen (wie hier) besonders gekennzeichnet.

TA1630

Darüberhinaus besitzt TA1630 COBOL zusätzliche Erweiterungen, auf die an den entsprechenden Stellen besonders hingewiesen wird und die durch Umrahmungen (wie hier gezeigt) besonders gekennzeichnet sind.

Vollständiges Sprachgerüst des TA 1600 - COBOL

Allgemeines Format der IDENTIFICATION DIVISION

IDENTIFICATION DIVISION.

PROGRAM-ID. Programmname

[AUTHOR. [Kommentar] ...]

[INSTALLATION. [Kommentar] ...]

[DATE-WRITTEN. [Kommentar] ...]

[SECURITY. [Kommentar] ...]

Allgemeines Format der ENVIRONMENT DIVISION

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  Computername  [WITH  DEBUGGING MODE].
OBJECT-COMPUTER.    Computername

      [, WITH NO SCREEN CONTROL]
      [, MEMORY SIZE Ganzzahl  { WORDS      } ]
                               { CHARACTERS }
                               { MODULES    }

      [, PROGRAM COLLATING SEQUENCE IS Alphabetname] .

[SPECIAL-NAMES.                    [[, Systemname

  { IS  Merkname                   [,ON STATUS IS Bedingungsname-1
                                   [,OFF STATUS IS Bedingungsname-2]
  { IS  Merkname                   [,OFF STATUS IS Bedingungsname-2
                                   [,ON STATUS IS Bedingungsname-1]]
  { ON STATUS IS Bed.name-1        [,OFF STATUS IS Bedingungsname-2]
  { OFF STATUS IS Bed.name-2       [,ON STATUS IS Bedingungsname-1]  } ]...

   [,Alphabetname  IS  { STANDARD-1 } ]  ...
                       { NATIVE     }

   [, CURRENCY SIGN IS Literal]
   [, DECIMAL-POINT IS COMMA] .]]

[INPUT-OUTPUT SECTION.
 FILE-CONTROL.

       {Dateisteuerungserklärung}

 [ I-O-CONTROL.
     [;SAME AREA FOR Dateiname-1, {Dateiname-2} ...]....
     [;MULTIPLE FILE TAPE CONTAINS
                Dateiname-3 [POSITION Ganzzahl-1]
            [,Dateiname-4 [POSITION Ganzzahl-2]] ...]...  ]]
```

Allgemeines Format der Dateisteuererklärung

Format 1

```
SELECT Dateiname

                  INPUT
                  INPUT-OUTPUT
ASSIGN TO         OUTPUT        ,nichtnumerisches Literal-1
                  PRINT         ,Datenname-0
                  RANDOM

[; RESERVE Ganzzahl-1  AREA ]
                       AREAS]
[;ORGANIZATION IS SEQUENTIAL]
[;ACCESS MODE IS SEQUENTIAL ]
[;FILE STATUS IS Datenname-1].
```

Format 2

```
SELECT  Dateiname

ASSIGN TO RANDOM  nichtnumerisches Literal
                  Datenname-0

[; RESERVE Ganzzahl-1  AREA ]
                       AREAS]

;ORGANIZATION IS RELATIVE
[;ACCESS MODE IS  SEQUENTIAL[,RELATIVE KEY IS Datenname-1]
                  RANDOM
                  DYNAMIC   ,RELATIVE KEY IS Datenname-1 ]
[;FILE STATUS IS Datenname-2]
```

Format 3

```
SELECT   Dateiname
ASSIGN TO RANDOM      nichtnumerisches Literal-1
                      Datenname-0

[;RESERVE Ganzzahl-1  AREA ]
                      AREAS ]
;ORGANIZATION IS INDEXED
[;ACCESS MODE IS  SEQUENTIAL
                  RANDOM
                  DYNAMIC ]
;RECORD KEY IS Datenname-1
[;ALTERNATE RECORD KEY IS Datenname-2 [ WITH DUPLICATES ] ] ...
[;FILE STATUS IS Datenname-3] .
```

TA TRIUMPH-ADLER
COB/Rev/1.32.d

Allgemeines Format der Dateisteuererklärung

TA 1620/TA 1630

Format 4

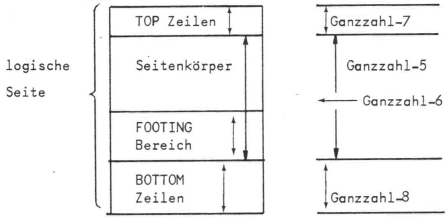
```
SELECT Dateiname
ASSIGN TO SCREEN                  {,nichtnumerisches Literal-1}
                                  {,Datenname-0               }

 [;RESERVE Ganzzahl-1             [AREA ]]
                                  [AREAS]]
 [;ORGANIZATION IS SEQUENTIAL]
 [;ACCESS MODE IS SEQUENTIAL]
 [;FILE STATUS IS Datenname-1]
 [;LINE AREA IS Ganzzahl-2  {THRU    }   Ganzzahl-3]
                            {THROUGH}
 [;SCREEN MODE IS           {SCROLL}]  .
                            {PAGE  }
```

Allgemeines Format der DATA DIVISION

```
DATA DIVISION.
[FILE SECTION.
[FD Dateiname
[;BLOCK CONTAINS [Ganzzahl-1 TO] Ganzzahl-2 {RECORDS / CHARACTERS}]
[;RECORD CONTAINS [Ganzzahl-3 TO] Ganzzahl-4 CHARACTERS]
;LABEL {RECORD / RECORDS} {IS / ARE} {STANDARD / OMITTED}
[;VALUE OF USER-LABEL IS {Datenname-1 / Literal-1}]
[;DATA {RECORD / RECORDS} {IS / ARE} Datenname-2 [,Datenname-3] ...]
[;LINAGE IS {Datenname-4 / Ganzzahl-5} LINES [,WITH FOOTING AT {Datenname-5 / Ganzzahl-6}]
[;LINES AT TOP {Datenname-6 / Ganzzahl-7}] [,LINES AT BOTTOM {Datenname-7 / Datenname-8}].
[ Satzbeschreibung ] ...] ...]
```

Die LINAGE-Klausel




```
[WORKING-STORAGE SECTION.
  [77-Datenbeschreibung
  Satzbeschreibung ]...]
LINKAGE SECTION.
  [77-Datenbeschreibung
  Satzbeschreibung ]...]
```

TA 1620/TA 1630

Bildschirm-Dateierklärung

FORMAT

```
FD Dateiname
 [;BLOCK CONTAINS  [ Ganzzahl-1 TO ]  Ganzzahl-2  {RECORDS    }]
                                                  {CHARACTERS}
 [;RECORD CONTAINS  [ Ganzzahl-3 TO ]  Ganzzahl-4  CHARACTERS]
  ;LABEL  {RECORD IS  }        {STANDARD}
          {RECORDS ARE}        {OMITTED }
 [;VALUE OF USER-LABEL IS      {Datenname-1}]
                               {Literal-1  }
 [;DATA  {RECORD IS  }
         {RECORDS ARE}    Datenname-3  [,Datenname-4]  ...]
 [;SCREEN IS Bildschirmsatzname-1    [,Bildschirmsatzname-2   ]...].
```

Allgemeines Format der Datenbeschreibung

FORMAT 1

```
Stufennummer {Datenname-1}
             {FILLER    }

[; REDEFINES  Datenname-2]

[; {PICTURE} IS Zeichenkette]
   {PIC    }

                 COMPUTATIONAL
                 COMP
                 COMPUTATIONAL-1
[;[USAGE IS] {   COMP-1         }]
                 COMPUTATIONAL-3
                 COMP-3
                 DISPLAY
                 INDEX

[;[SIGN IS] TRAILING  [SEPARATE CHARACTER]]
                      TA 1630
[;OCCURS  {Ganzzahl-1 TO Ganzzahl-2 TIMES}
          {   DEPENDING ON Datenname-3   }
          {Ganzzahl-2 TIMES              }

[ INDEXED BY  Indexname-1 [, Indexname-2] ...]

[; {SYNCHRONIZED} [LEFT ]]
   {SYNC        } [RIGHT]]

[; {JUSTIFIED} RIGHT]
   {JUST    }

[; BLANK WHEN ZERO ]

[; VALUE IS Literal].
```

FORMAT 2

```
66 Datenname-1 ;RENAMES Datenname-2
                   [{THROUGH} Datenname-3].
                    {THRU   }
```

FORMAT 3

```
88 Bedingungsname ; {VALUE  IS} Literal-1
                    {VALUES ARE}

                    [{THROUGH} Literal-2]
                     {THRU   }

                    [,Literal-3 [{THROUGH} Literal-4]]...
                                {THRU   }
```

TRIUMPH-ADLER 7A
CUR/Rec/1202.d

Allgemeines Format der Datenbeschreibung

FORMAT 4

Bildschirmdatenbeschreibung

```
level-number  { Datenname-1 }
              { FILLER      }
              { ATTRIBUTE   }

[; { PICTURE } IS Zeichenkette]
   { PIC     }

[;[USAGE IS]DISPLAY]
[;[SIGN IS] TRAILING [SEPARATE CHARACTER]]

[; { JUSTIFIED } RIGHT]
   { JUST      }

[; BLANK WHEN ZERO]
[; VALUE IS Literal]
[; LINE IS [PLUS] Ganzzahl-1]
[; POSITION IS [PLUS] Ganzzahl-2]
[; FIELD ATTRIBUTE IS

                    nicht TA1610
[, { [NOT OPTIONAL][AUTO][EXACT][WITH CHECK][NO ERASE] }]
   {                     PROTECTED                     }

[, { [INVERSE] [BLINKING] [BRIGHT] [UNDERLINED] }]
   {              NOT VISIBLE                   }

[, Merkname [,Merkname] ... ].
```

Allgemeines Format der PROCEDURE DIVISION

PROCEDURE DIVISION [USING Datenname-1 [,Datenname-2] ...].

FORMAT 1

[DECLARATIVES.

{Kapitelname SECTION [Segmentnummer] . Declarative-Satz
[Paragraphname. [Satz] ...] ... } ...
END DECLARATIVES.]

{Kapitelname SECTION [Segmentnummer].
[Paragraphname. [Satz] ...] ... } ...
[END PROGRAMM.]

FORMAT 2

{Paragraphname. [Satz] ... } ...

[END PROGRAM.]

Allgemeines Format der Verben

FORMAT 1

```
ACCEPT Bezeichner-1  [FROM Merkname-1]
 [DISPLAYING  [UPON Merkname-2] ]
 [LINE IS          {Bezeichner-2}]
                   {Literal-1    }
 [POSITION IS      {Bezeichner-3}]
                   {Literal-2    }
 [AFTER ERASE      [SCREEN]]
                   [LINE  ]
 [END KEY IS Merkname-3  [Merkname-4] ...
                 [POINTER IS Bezeichner-4]]
```

FORMAT 2

```
ACCEPT Bezeichner-1 FROM  {DATE}
                          {DAY }
                          {TIME}
```

ANMERKUNG

Die im folgenden beschriebene ACCEPT-Anweisung ist aus Kompatibilitätsgründen in TA 1600 COBOL mit aufgenommen worden und sollte für Programmentwicklungen nicht mehr verwendet werden.

FORMAT

```
ACCEPT Bezeichner-1  [,UNIT{Bezeichner-2}]  [,LINE{Bezeichner-3}]  [,POSITION {Bezeichner-4}]
                           {Literal-1   }          {Literal-2   }            {Literal-3   }
[SIZE{Bezeichner-5}][,PROMPT[Literal-5]][,ECHO][,CONVERT][,TAB][,NO BEEP][,ERASE]
     {Literal-4   }
[Bezeichner-6  [,UNIT {Bezeichner-7}]  [,LINE{Bezeichner-8}]  [,POSITION{Bezeichner-9}]
                      {Literal-6   }          {Literal-7   }            {Literal-8   }
[SIZE{Bezeichner-10}][,PROMPT[Literal-10]][,ECHO][,CONVERT][,TAB][,NO BEEP][,ERASE]]
     {Literal-9    }
```

FORMAT 1

```
ADD {Bezeichner-1} [,Bezeichner-2]  ... TO  Bezeichner-m [ROUNDED]
    {Literal-1   }  [,Literal-2  ]

    [; ON SIZE ERROR unbedingte Anweisung]
```

FORMAT 2

```
ADD {Bezeichner-1} , {Bezeichner-2} [,Bezeichner-3] ...
    {Literal-1   }   {Literal-2   } [,Literal-3   ]

    GIVING  Bezeichner-m [ROUNDED][; ON SIZE ERROR unbedingte Anweisung]
```

Allgemeines Format der Verben

TA 1630

FORMAT 3

```
ADD  { CORRESPONDING }  Bezeichner-1 TO Bezeichner-2 [ROUNDED]
     { CORR          }

     [;ON SIZE ERROR unbedingte Anweisung]
```

```
CALL  Literal-1  [USING  Datenname-1  [ ,Datenname-2 ] ... ]
```

FORMAT 1

```
CLOSE Dateiname-1 [ { REEL } [WITH NO REWIND] ] [,Dateiname-2 [ { REEL } [WITH NO REWIND] ] ]...
                    { UNIT }                                   { UNIT }
                    WITH { NO REWIND }                         WITH { NO REWIND }
                         { LOCK     }                              { LOCK     }
```

FORMAT 2

```
CLOSE  Dateiname-1  [WITH LOCK]  [ ,Dateiname-2 [WITH LOCK] ] ...
```

TA 1630

```
COMPUTE Bezeichner-1  [ROUNDED] = arithm. Ausdruck
    [;ON SIZE ERROR unbedingte Anweisung]
```

```
DELETE Dateiname RECORD  [;INVALID KEY unbedingte Anweisung]
```

FORMAT 1

```
DISPLAY  { Bezeichner-1 }  [UPON Merkname-1]
         { Literal-1    }

         [LINE IS        { Bezeichner-2 }]
                         { Literal-2    }

         [POSITION IS    { Bezeichner-3 }]
                         { Literal-3    }

         [AFTER ERASE    [ LINE   ]]  }...
                         [ SCREEN ]]  }
```

ANMERKUNG

Die im folgenden beschriebene DISPLAY-Anweisung ist aus Kompatibilitätsgründen in TA 1600 COBOL mit aufgenommen worden und sollte für Programmentwicklungen nicht mehr verwendet werden.

FORMAT 2

```
DISPLAY { Bezeichner-1 } [,UNIT { Bezeichner-2 }] [,LINE { Bezeichner-3 }]
        { Literal-1    }        { Literal-2    }         { Literal-3    }

[,POSITION { Bezeichner-4 }] [,SIZE { Bezeichner-5 }] [,BEEP] [,ERASE]
           { Literal-4    }         { Literal-5    }

[[ { Bezeichner-6 } [,UNIT { Bezeichner-7 }] [,LINE { Bezeichner-8 }]
   { Literal-6    }        { Literal-7    }         { Literal-8    }

[,POSITION { Bezeichner-9 }] [,SIZE { Bezeichner-10 }] [,BEEP][,ERASE]]]...
           { Literal-9    }         { Literal-10    }
```

Allgemeines Format der Verben

FORMAT 1

```
DIVIDE {Bezeichner-1} INTO Bezeichner-2 [ROUNDED]
       {Literal-1   }
 [; ON SIZE ERROR unbedingte Anweisung]
```

FORMAT 2

```
DIVIDE {Bezeichner-1} INTO {Bezeichner-2}
       {Literal-1   }      {Literal-2   }
GIVING Bezeichner-3 [ROUNDED][; ON SIZE ERROR unbedingte Anweisung]
```

FORMAT 3

```
DIVIDE {Bezeichner-1} BY {Bezeichner-2}
       {Literal-1   }    {Literal-2   }
GIVING Bezeichner-3 [ROUNDED][; ON SIZE ERROR unbedingte Anweisung]
```

EXIT.

EXIT PROGRAM.

TA 1620/TA 1630

```
GENERATE {Bildschirm-Satzname}
         {Satzname           }
```

GO TO[Prozedurname-1]

```
GO TO Prozedurname-1 [,Prozedurname-2] ..., Prozedurname-n
DEPENDING ON Bezeichner
```

```
IF Bedingung; {Anweisung-1  } [{; ELSE Anweisung-2  }]
              {NEXT SENTENCE} [{; ELSE NEXT SENTENCE}]
```

TA 1620/TA 1630

```
INITIATE Dateiname
```

Allgemeines Format der Verben

FORMAT 1

```
INSPECT Bezeichner-1 TALLYING

                        {ALL       }        {Bezeichner-3}
,Bezeichner-2 FOR ,     {LEADING   }        {Literal-1   }
                        {CHARACTERS}

  [{BEFORE} INITIAL {Bezeichner-4}]
   {AFTER }         {Literal-2   }

[REPLACING

{ CHARACTERS BY {Bezeichner-6} [{BEFORE} INITIAL {Bezeichner-7}]
                {Literal-4   }  {AFTER }         {Literal-5   }]

{ ,{ALL    }, {Bezeichner-5} BY {Bezeichner-6}
   {LEADING}  {Literal-3   }    {Literal-4   }
   {FIRST  }

  [{BEFORE} INITIAL {Bezeichner-7}]
   {AFTER }         {Literal-5   }]  .
```

FORMAT 2

```
INSPECT Bezeichner-1 REPLACING

{ CHARACTERS BY {Bezeichner-6} [{BEFORE} INITIAL {Bezeichner-7}]
                {Literal-4   }  {AFTER }         {Literal-5   }]

{ ,{ALL    }, {Bezeichner-5} BY {Bezeichner-6}
   {LEADING}  {Literal-3   }    {Literal-4   }
   {FIRST  }

  [{BEFORE} INITIAL {Bezeichner-7}]
   {AFTER }         {Literal-5   }]
```

Allgemeines Format der Verben

FORMAT 1

```
MOVE {Bezeichner-1} TO Bezeichner-2 [,Bezeichner-3] ...
     {Literal      }
```

TA 1630

FORMAT 2

```
MOVE {CORRESPONDING} Bezeichner-1 TO Bezeichner-2
     {CORR         }
```

FORMAT 3

```
MOVE {Bezeichner} TO RETURN VALUE
     {Literal   }
```

FORMAT 1

```
MULTIPLY {Bezeichner-1} BY Bezeichner-2 [ROUNDED]
         {Literal-1   }

 [; ON SIZE ERROR unbedingte Anweisung]
```

FORMAT 2

```
MULTIPLY {Bezeichner-1} BY {Bezeichner-2} GIVING Bezeichner-3 [ROUNDED]
         {Literal-1   }    {Literal-2   }

 [; ON SIZE ERROR unbedingte Anweisung]
```

```
                                                                  ..
OPEN {INPUT  Dateiname-1[WITH NO REWIND]  [,Dateiname-2 [WITH NO REWIND]]...}
     {OUTPUT Dateiname-3 [WITH NO REWIND][,Dateiname-4 [WITH NO REWIND]]...}
     {I-O    Dateiname-5                 [,Dateiname-6]                 ...}
     {EXTEND Dateiname-7                 [,Dateiname-8]                 ...}
```

Allgemeines Format der Verben

FORMAT 1

```
PERFORM  Prozedurname-1  [ { THROUGH } Prozedurname-2 ]
                           { THRU    }
```

FORMAT 2

```
PERFORM  Prozedurname-1  [ { THROUGH } Prozedurname-2 ]
                           { THRU    }

{ Bezeichner-1 } TIMES
{ Ganzzahl     }
```

FORMAT 3

```
PERFORM  Prozedurname-1  [ { THROUGH } Prozedurname-2 ]
                           { THRU    }

  UNTIL  Bedingung-1
```

FORMAT 4

```
PERFORM  Prozedurname-1  [ { THROUGH } Prozedurname-2 ]
                           { THRU    }

     VARYING { Bezeichner-2 } FROM { Bezeichner-3 }
             { Indexname-1  }      { Indexname-2  }
                                   { Literal-1    }

        BY { Bezeichner-4 } UNTIL Bedingung-1
           { Literal-2    }

  [ AFTER { Bezeichner-5 } FROM { Bezeichner-6 }
          { Indexname-3  }      { Indexname-4  }
                                { Literal-3    }

        BY { Bezeichner-7 } UNTIL Bedingung-2
           { Literal-4    }

  [ AFTER { Bezeichner-8 } FROM { Bezeichner-9 }
          { Indexname-5  }      { Indexname-6  }
                                { Literal-5    }

        BY { Bezeichner-10 } UNTIL Bedingung-3 ] ]
           { Literal-6     }
```

Allgemeines Format der Verben

FORMAT 1

```
READ Dateiname [NEXT] RECORD [WITH NO LOCK] [INTO Bezeichner]
 [;AT END unbedingte Anweisung]
```

FORMAT 2

```
READ Dateiname RECORD [WITH NO LOCK] [INTO Bezeichner]
 [; KEY IS Datenname]
 [;INVALID KEY unbedingte Anweisung]
```

FORMAT 3 (SCREEN I/O)

TA 1620/TA 1630

```
READ Dateiname RECORD [WITH MASK] [INTO Bezeichner-1]
    [;END KEY IS Merkname-1   [,Merkname-2] ...
        [POINTER IS Bezeichner-2]]
    [;AT END unbedingte Anweisung]
```

```
REWRITE  Satzname  [FROM Bezeichner]
 [;INVALID KEY unbedingte Anweisung]
```

FORMAT 1

```
SET {Bezeichner-1 [,Bezeichner-2] ...} TO {Bezeichner-3}
    {Indexname-1  [,Indexname-2 ] ...}    {Indexname-3}
                                          {Ganzzahl-1 }
```

FORMAT 2

```
SET Indexname-4 [,Indexname-5] ... {UP BY}  {Bezeichner-4}
                                   {DOWN BY}{Ganzzahl-2  }
```

```
START Dateiname [ KEY {IS EQUAL TO   } Datenname ]
                      {IS=           }
                      {IS GREATER THAN}
                      {IS >          }
                      {IS NOT LESS THAN}
                      {IS NOT <      }
 [;INVALID KEY unbedingte Anweisung]
```

```
STOP {RUN    }
     {Literal}
```

Allgemeines Format der Verben

FORMAT 1

```
SUBTRACT  {Bezeichner-1} [, Bezeichner-2] ...
          {Literal-1   } [, Literal-2   ]

FROM   Bezeichner-m [ROUNDED][; ON SIZE ERROR unbedingte Anweisung ]
```

FORMAT 2 (SCREEN I/O)

```
SUBTRACT  {Bezeichner-1} [, Bezeichner-2] ... FROM {Bezeichner-m}
          {Literal-1   } [, Literal-2   ]          {Literal-m   }

GIVING  Bezeichner-n [ROUNDED][; ON SIZE ERROR unbedingte Anweisung]
UNLOCK  Dateiname  RECORD
```

TA 1630

```
FORMAT 3
SUBSTRACT {CORRESPONDING}
          {CORR          }
        Bezeichner-1 FROM Bezeichner-2 [ROUNDED]
        [; ON SIZE ERROR unbedingte Anweisung]
```

FORMAT 1

```
USE AFTER STANDARD {EXCEPTION} PROCEDURE ON {Dateiname-1}
                   {ERROR    }             {INPUT      }
                                           {OUTPUT     }
                                           {I-O        }
                                           {EXTEND     }
                                           {SCREEN     }
```

FORMAT 2

TA 1620/TA 1630

```
USE FOR CHECK ON Dateiname-1
```

FORMAT 1

```
WRITE Satzname  [FROM Bezeichner-1]
[ {BEFORE} ADVANCING {Bezeichner-2} [LINE ]  {Bezeichner-3} ]
  {AFTER }           {Ganzzahl    } [LINES]  {Ganzzahl-2   }
                     TO LINE
                     PAGE
[; AT {END-OF-PAGE} unbedingte Anweisung]
      {EOP        }
```

FORMAT 2

```
WRITE Satzname  [FROM Bezeichner]  [;INVALID KEY unbedingte Anweisung]
```

Allgemeines Format der Bedingungen

Vergleichsbedingungen

```
{Bezeichner-1}  {IS [NOT]GREATER THAN}  {Bezeichner-2}
{Literal-1   }  {IS [NOT]LESS THAN    }  {Literal-2  }
                {IS [NOT]EQUAL TO     }  {Indexname-2}
                {IS [NOT] >           }
                {IS [NOT] <           }
                {IS [NOT] =           }
```

Klassenbedingung

```
Bezeichner IS [NOT] {NUMERIC   }
                    {ALPHABETIC}
```

Schalterzustandsbedingung

Bedingungsname

Zusammengesetzte Bedingungen

```
Bedingung {AND} Bedingung ...
          {OR}
```

Verschiedene Formate

Qualifizierung

```
{Datenname-1   }{[{OF}  Datenname-2] ...
{Bedingungsname}{[{IN}
Paragraphenname [{OF}  Kapitelname]
                [{IN}
```

Subskribierung

Datenname (Subskript-1 ,[Subskript-2 [,Subskript-3]])

Indizierung

```
Datenname ({Indexname-1  [{+} Literal-2]}
           {Literal-1                   }
  [{Indexname-2  [{+} Literal-4]}  [{Indexname-3  [{+} Literal-6]}]])
  [{Literal-3                   }  [{Literal-5                   }
```

Allgemeines Format der Bedingungen

Bezeichner

FORMAT 1

```
Datenname-1  [ { OF } Datenname-2 ] ... [(Subskript-1[,Subskript-2[,Subskript-3]])]
               { IN }
```

FORMAT 2

```
Datenname-1  [ { OF } Datenname-2 ] ... [( { Indexname-1 [ { + } Literal-2] } )]
               { IN }                    { Literal-1     { - }            }
[ { Indexname-2 [ { + } Literal-4] } [ { Indexname-3 [ { + } Literal-6] } ] ]
  { Literal-3     { - }            } { Literal-5     { - }            }
```

Allgemeines Format der COPY-Anweisung

COPY Textname

Reservierte Wörter

ACCEPT	DATA	GENERATE	MODULES
ACCESS	DATE	GIVING	MOVE
ADD	DATE-WRITTEN	GO	MULTIPLE
ADVANCING	DAY	GREATER	MULTIPLY
AFTER	DEBUGGING		
ALL	DECIMAL-POINT	HIGH-VALUE	NATIVE
ALPHABETIC	DECLARATIVES	HIGH-VALUES	NEXT
ALTERNATE	DELETE		NO
AND	DEPENDING	I-O	NOT
ARE	DISPLAY	I-O-CONTROL	NUMERIC
AREA	DISPLAYING	IDENTIFICATION	
AREAS	DIVIDE	IF	OBJECT-COMPUTER
ASSIGN	DIVISION	IN	OCCURS
AT	DOWN	INDEX	OF
ATTRIBUTE	DUPLICATES	INDEXED	OFF
AUTHOR	DYNAMIC	INITIAL	OMITTED
AUTO		INITIATE	ON
BEFORE	ECHO	INPUT	OPEN
BLANK	ELSE	INPUT-OUTPUT	OPTIONAL
BLINKING	END	INSPECT	OR
BLOCK	END-OF-PAGE	INSTALLATION	ORGANIZATION
BOTTOM	ENVIRONMENT	INTO	OUTPUT
BRIGHT	EOP	INVALID	
BY	EQUAL	INVERSE	PAGE
CALL	ERASE	IS	PERFORM
CANCEL **	ERROR		PIC
CHARACTER	EXACT	JUST	PICTURE
CHARACTERS	EXIT	JUSTIFIED	PLUS
CHECK	EXTEND		POINTER
CHECK-ITEM			
CHECK-RESPONSE			
CHECK-MESSAGE			
CLOSE		KEY	POSITION
COLLATING	F1		PRINT
COMMA	F2	LABEL	PROCEDURE
COMP	F3	LEADING	PROCEED
COMP-1	F4	LEFT	PROGRAM
COMP-3	F5	LESS	PROGRAM-ID
COMPUTATIONAL	F6	LINAGE	PROMPT
COMPUTATIONAL-1	F7	LINAGE-COUNTER	PROTECTED
COMPUTATIONAL-3	F8	LINE	
COMPUTE*	FD	LINES	QUOTE
CONFIGURATION	FIELD	LINKAGE	QUOTES
CONTAINS	FILE	LOCK	
CONTROL	FILE-CONTROL	LOW-VALUE	RANDOM
CONVERT	FILLER	LOW-VALUES	READ
COPY	FIRST		RECORD
CORR*	FOOTING	MASK	RECORDS
CORRESPONDING*	FOR	MEMORY	REDEFINES
CRT-STATUS	FROM	MODE	REEL
CURRENCY			

* Wörter, die nur in TA 1630 COBOL verwendet werden.
** Wörter, die nur in TA 1620/TA 1610 verwendet werden.

Reservierte Wörter (Fortsetzung)

RELATIVE
REMOVAL
RENAMES
REPLACING
RESERVE
RETURN
RETURN-KEY
REWRITE
RIGHT
ROUNDED
RUN
SAME
SCREEN
SCROLL
SECTION
SECURITY
SELECT
SEND-KEY
SENTENCE
SEPARATE
SEQUENCE
SEQUENTIAL
SET
SIGN
SIZE
SOURCE-COMPUTER
SPACE
SPACES
SPECIAL-NAMES
ST01
ST02
ST03
ST04
ST05
ST06
ST07
ST08
ST09
ST10
ST11
ST12
ST13
ST14
ST15
STANDARD
STANDARD-1
START
STATUS
STOP
SUBTRACT
SYNC
SYNCHRONIZED
SWITCH-1
SWITCH-2
SWITCH-3
SWITCH-4
SWITCH-5
SWITCH-6
SWITCH-7
SWITCH-8
TALLYING
TAPE
THAN
THEN
THROUGH
THRU
TIME
TIMES
TO
TOP
TRAILING
UNIT
UNDERLINED
UNLOCK
UNTIL
UP
UPON
USAGE
USE
USER-LABEL
VALUE
VALUES
VARYING
VISIBLE
WHEN
WITH
WORDS
WORKING-STORAGE
WRITE
ZERO
ZEROES
ZEROS
+
-
/
>
<
=

Zusätzlich im ANSI reservierte Wörter

ALSO
ALTER
ASCENDING

CANCEL
CD
CF
CH
CLOCK-UNITS
COBOL
CODE
CODE-SET
COLUMN
COMMUNICATION
CONTROLS
COUNT

DATE-COMPILED
DE
DEBUG-CONTENTS
DEBUG-ITEM
DEBUG-LINE
DEBUG-NAME
DEBUG-SUB-1
DEBUG-SUB-2
DEBUG-SUB-3
DELIMITED
DELIMITER
DESCENDING
DESTINATION
DETAIL
DISABLE

EGI
EMI
ENABLE
ENTER
ESI
EVERY

FINAL

GROUP

HEADING

INDICATE

LAST
LENGTH
LIMIT
LIMITS
LINE-COUNTER

MERGE
MESSAGE

NEGATIVE
NUMBER

OVERFLOW

PAGE-COUNTER
PF
PH
POINTER
POSITIVE
PRINTING
PROCEDURES
PROCEED

QUEUE

RD
RECEIVE
REFERENCES
RELEASE
REMAINDER
REPORT
REPORTING
REPORTS
RERUN
RESET
REVERSED
RF
RH

SD
SEARCH
SEGMENT
SEGMENT-LIMIT
SEND
SORT
SORT-MERGE
SOURCE
STRING
SUB-QUEUE-1
SUB-QUEUE-2
SUB-QUEUE-3
SUM
SUPPRESS
SYMBOLIC

TABLE
TERMINAL
TERMINATE
TEXT
TYPE

UNSTRING

Vorrangregeln für PICTURE-Zeichen

2. Zeichen	1. Zeichen	Feste Einfügung									Gleitende Einfügung						Andere Symbole					
		B	O	/	,	.	{+ −}	{+ −}	{CR DB}	CS	{Z *}	{Z *}	{+ −}	{+ −}	CS	CS	9	A X	S	V	P	P
Feste Einfügung	B	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X.		X		X
	O	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X		X		X
	/	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X		X		X
	,	X	X	X	X	X	X			X	X	X	X	X	X	X	X			X		X
	.	X	X	X	X		X			X	X		X		X		X					
	{+ −}																					
	{+ −}	X	X	X	X	X				X	X	X			X	X	X			X	X	X
	{CR DB}	X	X	X	X	X				X	X	X			X	X	X			X	X	X
	CS						X															
Gleitende Einfügung	{Z *}	X	X	X	X		X			X	X											
	{Z *}	X	X	X	X	X	X			X	X	X								X		X
	{+ −}	X	X	X	X					X			X									
	{+ −}	X	X	X	X	X				X			X	X						X		X
	CS	X	X	X	X		X								X							
	CS	X	X	X	X	X	X								X	X				X		X
Andere Symbole	9	X	X	X	X	X	X			X	X		X		X		X	X	X	X		X
	A X	X	X	X													X	X				
	S																					
	V	X	X	X	X		X			X	X		X		X		X		X		X	
	P	X	X	X	X		X			X	X		X		X		X		X		X	
	P						X			X									X	X		X

Klassen und Kategorien der Datenfelder

Stufe des Datenfeldes	Klasse	Kategorie
Elementar	Alphabetisch	Alphabetisch
	Numerisch	Numerisch
	Alphanumerisch	Numerisch-druckaufbereitet
		Alphanumerisch-druckaufbereitet
		Alphanumerisch
Nichtelementar (Gruppe)	Alphanumerisch	Alphabetisch
		Numerisch
		Numerisch-druckaufbereitet
		Alphanumerisch-druckaufbereitet
		Alphanumerisch

Erlaubte Operanden der MOVE-Anweisung

Kategorie des Sendefeldes	alpha-betisch	Kategorie des Empfangsfeldes alphanumerisch druckaufbereitet alphanumerisch	numerische Ganzzahl numerische Nicht-Ganzzahl numerisch-druckaufbereitet
alphabetisch	Ja	Ja	Nein
alphanumerisch	Ja	Ja	Ja
alphanum.-druck-aufbereitet	Ja	Ja	Nein
numerische Ganz-zahl	Nein	Ja	Ja
numerische Nicht-Ganzzahl	Nein	Nein	Ja
numerisch-druck-aufbereitet	Nein	Ja	Ja *

* Ausnahme: Numerisch-druckaufbereitetes Empfangsfeld

Kombination der Operanden in der SET-Anweisung

Sendefeld	Empfangsfeld Ganzzahliges Datenfeld	Indexname	Indexdatenname
ganzzahliges Literal	NEIN	GÜLTIG	NEIN
ganzzahliges Datenfeld	NEIN	GÜLTIG	NEIN
Indexname	GÜLTIG	GÜLTIG	GÜLTIG *
Indexname	NEIN	GÜLTIG *	GÜLTIG *

* Es wird keine Konvertierung durchgeführt

TA TRIUMPH-ADLER
COB/Ref/1202/A

Dateiorganisation

Bedeutung der Zustandsschlüssel für sequentielle Dateien

Zustands-schlüssel 1	Zustands-schlüssel 2	Bedeutung
0	0	Erfolgreiche Beendigung
1	0	AT END-Bedingung
3	0	Permanenter Fehler
3	4	Permanenter Fehler, Überschreitung der Dateigrenzen
9	0	Vom Betriebssystem erkannter Fenler, ungültige Operation
9	1	Vom Betriebssystem erkannter Fehler, Datei nicht eröffnet
9	2	Vom Betriebssystem erkannter Fehler, Datei nicht geschlossen
9	3	Vom Betriebssystem erkannter Fehler, Datei nicht verfügbar
9	4	Vom Betriebssystem erkannter Fehler, ungültige OPEN-Funktion
9	5	Vom Betriebssystem erkannter Fehler, ungültiges Gerät
9	6	Vom Betriebssystem erkannter Fehler, ungültiger aktueller Satzzeiger
9	7	Vom Betriebssystem erkannter Fehler, ungültige Satzlänge
9	9	Vom Betriebssystem erkannter Fehler, gesperrter (locked) Satz

Zulässige Anweisungen (sequentiell)

	OPEN-Modus			
Anweisung	INPUT	OUTPUT	INPUT-OUTPUT	EXTEND
READ	X		X	
WRITE		X		X
REWRITE			X	

Bedeutung der Zustandsschlüssel für Indexsequentielle Dateien

Zustands-schlüssel-1	Zustands-schlüssel-2	Bedeutung
0	0	Erfolgreiche Beendigung
0	2	Erfolgreiche Beendigung; Duplikat erkannt
1	0	AT END-Bedingung
2	1	INVALID KEY-Bedingung; Reihenfolgefehler bei sequentiellem Zugriff auf Indexdateien
2	2	INVALID KEY-Bedingung; doppelt vorhandener KEY-Wert
2	3	INVALID KEY-Bedingung; kein Satz gefunden
2	4	INVALID KEY-Bedingung; Überschreiten der Dateigrenzen
3	0	Permanenter Fehler
3	4	Permanenter Fehler, Überschreitung der Dateigrenzen
9	0	Vom Betriebssystem erkannter Fehler, ungültige Operation
9	1	Vom Betriebssystem erkannter Fehler, Datei nicht eröffnet
9	2	Vom Betriebssystem erkannter Fehler, Datei nicht geschlossen
9	3	Vom Betriebssystem erkannter Fehler, Datei nicht verfügbar
9	4	Vom Betriebssystem erkannter Fehler, ungültige OPEN-Funktion
9	5	Vom Betriebssystem erkannter Fehler, ungültiges Gerät
9	6	Vom Betriebssystem erkannter Fehler, undefinierter interner Satzzeiger
9	7	Vom Betriebssystem erkannter Fehler, ungültige Satzlänge
9	8	Vom Betriebssystem erkannter Fehler, ungültiger Index
9	9	Vom Betriebssystem erkannter Fehler, gesperrter (locked) Satz

Tabelle der zulässigen Anweisungen (indexsequentiell)

		OPEN - Modus		
Dateizugriffsmodus	Anweisung	INPUT	OUTPUT	INPUT-OUTPUT
SEQUENTIAL	READ	X		X
	WRITE		X	
	REWRITE			X
	START	X		X
	DELETE			X
RANDOM	READ	X		X
	WRITE		X	X
	REWRITE			X
	START			
	DELETE			X
DYNAMIC	READ	X		X
	WRITE		X	X
	REWRITE			X
	START	X		X
	DELETE			X

Bedeutung der Zustandsschlüssel für relative Dateien

Zustands-schlüssel-1	Zustands-schlüssel-2	Bedeutung
0	0	Erfolgreiche Beendigung
1	0	AT END-Bedingung
2	2	INVALID KEY-Bedingung; doppelt vorhandener KEY-Wert
2	3	INVALID KEY-Bedingung; kein Satz gefunden
2	4	INVALID KEY-Bedingung; Überschreiten der Dateigrenzen
3	0	Permanenter Fehler
3	4	Permanenter Fehler, Überschreitung der Dateigrenzen
9	0	Vom Betriebssystem erkannter Fehler, ungültige Operation
9	1	Vom Betriebssystem erkannter Fehler, Datei nicht eröffnet
9	2	Vom Betriebssystem erkannter Fehler, Datei nicht geschlossen
9	3	Vom Betriebssystem erkannter Fehler, Datei nicht verfügbar
9	4	Vom Betriebssystem erkannter Fehler, ungültige OPEN-Funktion
9	5	Vom Betriebssystem erkannter Fehler, ungültiges Gerät
9	6	Vom Betriebssystem erkannter Fehler, undefinierter interner Satzzeiger
9	7	Vom Betriebssystem erkannter Fehler, ungültige Satzlänge
9	9	Vom Betriebssystem erkannter Fehler, gesperrter (locked) Satz

Tabelle der zulässigen Anweisungen (relativ)

		OPEN - Modus		
Dateizugriffsmodus	Anweisung	INPUT	OUTPUT	INPUT-OUTPUT
SEQUENTIAL	READ	X		X
	WRITE		X	
	REWRITE			X
	START	X		X
	DELETE			X
RANDOM	READ	X		X
	WRITE		X	X
	REWRITE			X
	START			
	DELETE			X
DYNAMIC	READ	X		X
	WRITE		X	X
	REWRITE			X
	START	X		X
	DELETE			X

Bedeutung der Zustandsschlüssel für SCREEN-Verarbeitung

Zustands-schlüssel 1	Zustands-schlüssel 2	Bedeutung
0	0	Erfolgreiche Beendigung
1	0	AT END-Bedingung
3	0	Permanenter Fehler
3	9	Permanenter Fehler, System-Restart
9	0	Vom Betriebssystem erkannter Fehler, ungültige Operation
9	1	Vom Betriebssystem erkannter Fehler, Datei nicht eröffnet
9	2	Vom Betriebssystem erkannter Fehler, Datei nicht geschlossen
9	3	Vom Betriebssystem erkannter Fehler, Datei nicht verfügbar
9	4	Vom Betriebssystem erkannter Fehler, ungültige OPEN-Funktion
9	5	Vom Betriebssystem erkannter Fehler, ungültiges Gerät

Tabelle der erlaubten I/O-Befehle (Bildschirm-Modus)

		OPEN MODUS		
SCREEN MODUS	OPERATION	INPUT	OUTPUT	I-O
	READ	X		X
S	WRITE		X	
C	DELETE			X
R	GENERATE		X	X
O	INITIATE	X		X
L				
L				
	READ	X		X
P	WRITE		X	X
A	DELETE			X
G	GENERATE		X	X
E	INITIATE	X		X

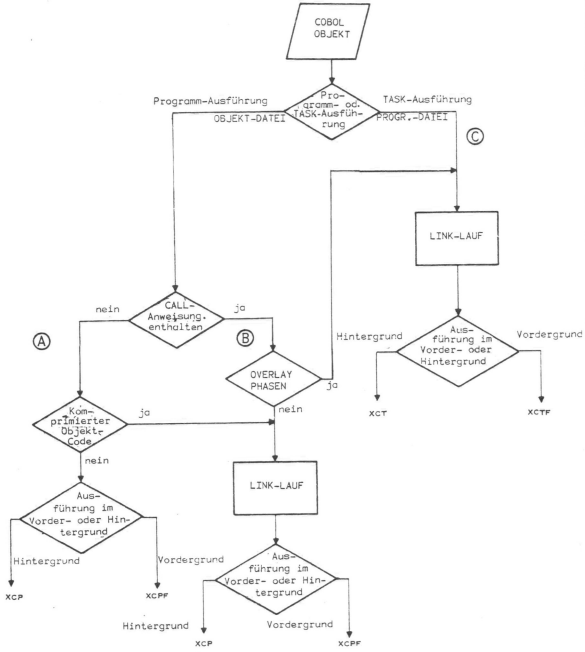
Bedienung des TA 1630 COBOL-Systems

Ausführung des COBOL-Compilers im Vordergrund
(Execute COBOL Compiler in Foreground - XCCF)

```
EXECUTE TA1630 COBOL COMPILER IN FOREGROUND <VERSION i.i.i.>
    SOURCE ACCESS NAME:
    OBJECT ACCESS NAME:
   LISTING ACCESS NAME:
           OPTIONS:
       PRINT WIDTH: 132
         PAGE SIZE: 55
 PROGRAM SIZE (LINES): 1000
   COPYLIB ACCESS NAME:
```

Viii gibt dabei die Versionsnummer des Compilers an.

Flußdiagramm zur Bestimmung, ob Linken für COBOL-Programm erforderlich



XCP - EXECUTE COBOL PROGRAM BACKGROUND
(AUSFÜHRUNG EINES COBOL-PROGRAMMES IM HINTERGRUND)

XCPF - EXECUTE COBOL PROGRAM FOREGROUND
(AUSFÜHRUNG EINES COBOL-PROGRAMMES IM VORDERGRUND)

XCT - EXECUTE COBOL TASK BACKGROUND
(AUSFÜHRUNG EINER COBOL-TASK IM HINTERGRUND)

XCTF - EXECUTE COBOL TASK FOREGROUND
(AUSFÜHRUNG EINER COBOL-TASK IM VORDERGRUND)

Wahlfreie Angaben für den COBOL-Compiler

Angabe	Bedeutung
D	Debug
O	Objektliste
X	Crossreferenzliste
R	ROM/RAM-Compilierung
E	ohne Formatierung

Ausführung des COBOL-Compilers im Hintergrund
(Execute COBOL Compiler in Background - XCC)

```
EXECUTE TA1630 COBOL COMPILER < VERSION i.i.i.> FOREGROUND
    SOURCE ACCESS NAME:
    OBJECT ACCESS NAME:
   LISTING ACCESS NAME:
            OPTIONS:
        PRINT WIDTH: 132
          PAGE SIZE: 55
  PROGRAM SIZE (LINES): 6000
   COPYLIB ACCESS NAME:
```

Die Parameter sind dieselben wie beim XCCF-Kommando, mit der Ausnahme, daß bei LISTING ACCESS NAME nicht ME angegeben werden darf.

Ausführung eines COBOL-Programmes im Vordergrund
(Execute COBOL Program in Foreground - XCPF)

```
EXECUTE TA1630 COBOL PROGRAM IN FOREGROUND < VERSION i.i.i.>
 OBJECT ACCESS NAME :
         DEBUG MODE :  NO
MESSAGE ACCESS NAME :
           SWITCHES :  00000000
```

Ausführung eines COBOL-Programmes im Hintergrund
(Execute COBOL Program in Background - XCP)

```
EXECUTE TA1630 COBOL PROGRAM <VERSION i.i.i.>
 OBJECT ACCESS NAME :
         DEBUG MODE : NO
MESSAGE ACCESS NAME :
           SWITCHES : 00000000
```

Erlaubte Link-Editor-Anweisungen bei COBOL-Objektprogrammen

Anweisungen (die unterstrichenen Anweisungen werden standardmäßig als geschrieben angenommen	Teilbinden (PARTIAL)	Ausführung (einer Objektdatei)	Ausführung (einer Programmdatei)
ADJUST	Y	Y	Y
ALLOCATE	NO	NO	Y
AUTO	(1)	Y	Y
COMMON	NO	NO	NO
DATA	NO	NO	NO
DUMMY	Y	NO	Y
END	Y	Y	Y
ERROR/NO ERROR	Y	Y	Y
FORMAT ASCII	Y	Y	NO
FORMAT COMPRESSED	Y	Y	NO
FORMAT IMAGE	NO	NO	Y
FORMAT IMAGE, REPLACE	NO	NO	Y
GLOBAL/ALL GLOBAL/NOT GLOBAL	Y	NO	NO
INCLUDE	Y	(2)	(3)
LIBRARY	Y	Y	Y
LOAD/NO LOAD	NO	NO	Y
MAP/NO MAP	Y	Y	Y
MEMORY			
NOAUTO	Y	Y	Y
NOSYMT	Y	Y	Y
PAGE/NOPAGE	Y	Y	Y
PARTIAL	Y	NO	NO
PHASE 0	Y	Y	Y
PHASE 1,2, . . . n	NO	NO	Y
PROCEDURE	NO	NO	Y
PROGRAM	NO	NO	NO
SEARCH	Y	Y	Y
SHARE	NO	NO	NO
SYMT	Y	Y	NO
TASK	Y	Y	Y

Anmerkungen

Y - Anweisung ist erlaubt
NO- Anweisung ist nicht erlaubt
(1) Bei Teilbinden (PARTIAL) wird standardmäßig NOAUTO als geschrieben angenommen; diese Anweisungen sollten dann nicht angegeben werden.
(2) Das Hauptprogramm muß bei der ersten INCLUDE-Anweisung stehen.
(3) Die Module des COBOL-Laufzeitsystems RCBTSK, RCBPRC und RCBMPD müssen durch diese Anweisungen in den Link-Lauf mit einbezogen werden.

COBOL-Laufzeitsystem

Das COBOL-Laufzeitsystem besteht aus drei vorgelinkten Modulen:

.S$SYSLIB.RCBTSK - Dies ist der Task-Teil des COBOL-Laufzeitinterpreters. Durch die INCLUDE-Anweisung muß er als erstes Modul in den Task-Teil der Ausgabe des Link-Editors mit einbezogen werden. RCBTSK ist nicht reentrant.

.S$SYSLIB.RCBPRC - Dies ist der reentrante Teil des COBOL-Laufzeitinterpreters. Falls gewünscht, kann er durch die INCLUDE-Anweisung als eine Prozedur mit einbezogen werden.

.S$SYSLIB.RCBMPD - Dieses Modul bestimmt, welches COBOL-Objektmodul das Hauptprogramm sein soll. Dazu muß dieses Modul des Laufzeitinterpreters direkt vor dem entsprechenden COBOL-Objektmodul mit einbezogen werden. Da RCBMPD reentrant ist, kann es auch in Prozeduren verwendet werden.

XCTF-Kommando

Mit dem XCTF-Kommando wird eine COBOL-Task, die zuvor auf einer Programmdatei installiert wurde, im Vordergrund ausgeführt.
Das Format des Kommandos stellt sich folgendermaßen dar:

```
EXECUTE COBOL TASK FOREGROUND  <VERSION  x.x.x.>

    PROGRAM FILE LUNO:
      TASK ID OR NAME:
                DEBUG: NO
  MESSAGE ACCESS NAME:
            SWITCHES: 00000000
```

XCT-Kommando

Mit dem XCT-Kommando wird eine COBOL-Task, die zuvor auf einer Programmdatei installiert wurde, im Hintergrund ausgeführt.
Das Format des Kommandos stellt sich folgendermaßen dar:

```
EXECUTE COBOL TASK  <VERSION  x.x.x.>

    PROGRAM FILE LUNO:
      TASK ID OR NAME:
                DEBUG: NO
  MESSAGE ACCESS NAME:
            SWITCHES: 00000000
```

Ausführungscodes von COBOL-Programmen

Wird ein COBOL-Programm über eine Kommando-Prozedur ausgeführt, so wird in dem Synonym $$CC ein Ausführungscode abgelegt.
$$CC kann dabei folgende Werte annehmen:

0000 - Normale Beendigung
8000 - Fehlerhafte Beendigung

Mit der MOVE - TO RETURN-Value - Anweisung können den 2 rechten Stellen des Ausführungscodes beliebige Werte zugewiesen werden.

Bedienung des TA 1620 COBOL-Systems

Compilierung an der TA 1620 (TA 1620 COBOL-Compiler)

Der COBOL-Compiler wird aktiviert, indem man das JDL-Kommando COBOL ausführt; dieses Kommando hat folgendes Format:

/COBOL,OPTIONS=*(NAME) [DEBUG,NL,NO,XREF,OL,NF,TR,TA1610]

Der Compiler liest das COBOL-Quelldeck von der Datei SI, gibt die Compilerliste auf die Datei LO und den binären Objektcode auf die Datei BO aus.

Die oben erwähnten Dateien müssen vor Ausführung des COBOL-Compilers (mit dem JDL-Kommando CREATE) angelegt und (mit dem JDL-Kommando ASSIGN) den logischen Namen SI, LO und BO zugewiesen worden sein.

Die Compileroptions haben folgende Bedeutung:

DEBUG	=	Die Debug-Zeilen (D in Spalte 7) werden mit übersetzt
NL	=	Unterdrückung der Compilerliste
NO	=	Unterdrückung der Objektcodeausgabe
XREF	=	Ausgabe einer Crossreferenzliste
OL	=	Ausgabe einer Compilerliste mit Objektcode
NF	=	Bildschirm I/O wird nicht unterstützt
TR	=	Trace (nicht implementiert)
TA1610	=	Prüfung auf TA 1610 Sprachumfang

Compilierung an der TA 1630 (Crosscompiler)

Der COBOL CROSSCOMPILER läuft auf dem System TA 1630 und erzeugt Objektcode für das System TASO. Er besteht aus dem Cobolcompiler des Systems TASO und IO-Routinen des System TA 1630. Der Crosscompiler liest das Quelldeck von einer TA 1630 Datei und generiert TASO Objektcode in eine TA 1630 Datei. Der Code wird mit einem Dienstprogramm in eine TASO Programmdatei überspielt und kann nun auf der TA 1610 oder TA 1620 ausgeführt werden.

Programm-Compilierung

Der COBOL-Crosscompiler wird aktiviert, indem man die Kommandos XCCRC (EXECUTE COBOL CROSS COMPILER) oder XCCRCF (EXECUTE COBOL CROSS COMPILER IN FOREGROUND) eingibt.

```
EXECUTE TA 1600 COBOL CROSS COMPILER Viii FOREGROUND

 SOURCE ACCESS NAME:
 OBJECT ACCESS NAME:
LISTING ACCESS NAME:
            OPTIONS:
        PRINT WIDTH: 132
          PAGE SIZE: 55
PROGRAM SIZE(LINES): 6000
COPYLIB ACCESS NAME:
```

COBOL-OPTION IM JDL-KOMMANDO	CROSSCOMPILER-OPTION
DEBUG	D
OL	O
NL	N
XREF	X
NO,TR	nicht unterstützt
NF	E
TA1610	A

Transfer des Objektcodes

```
TRANSFER CROSSCOMPILER OBJEKT
    DISKETTE ACCESS NAME:
          TAXO FILE NAME:
          TASO FILE NAME:
                RELACE?: NO
```

Compilierung und Transfer mit dem Kommando XCCRCT(F)

```
EXECUTE COBOL CROSS COMPILER Viii WITH TRANSFER
     SOURCE ACCESS NAME:
   DISKETTE ACCESS NAME:
       TASO ACCESS NAME:
                REPLACE?:
    LISTING ACCESS NAME:
               OPTIONS:
            PRINT WIDTH:  132
              PAGE SIZE:   55
    PROGRAM SIZE(LINES): 6ØØØ
    COPYLIB ACCESS NAME:
```

Ausführung von COBOL Programmen an der TA 1620

Nachdem der binäre Objektcode in eine TA 1620 Programmdatei gebracht wurde, kann das Programm an der TA 1620 mit dem JDL-Kommando EXECUTE ausgeführt werden.
Dieses Kommando hat das folgende Format:

```
/EXECUTE, LOGICAL NAME = NAME,
          OPTIONS =*NAME(DEBUG)
          SWAPPING PARTITION = *INT,
          PARAMETERS = *(INT)
```

COBOL-Laufzeit-Debug

Wird die Standardeintragung "NO" gewählt, dann läuft das Programm normal ab.
Wird jedoch "YES" eingetragen, wird das Programm unter der Kontrolle des Debug-Programmes gestartet.

Debug-Kommandos

Kommando	Name
B	Stop Adresse
S	Single
D	Dump
K	Beende Programmausführung (Kill)
N	Beende Debug-Modus (No Debug)

In der Programmliste wird für jedes Datenfeld sowohl der Datentyp als auch die entsprechende Abkürzung ausgegeben.

Die Abkürzungen sind:

NBS - Binär mit Vorzeichen (COMP-1)
NPS - Numerisch gepackt mit Vorzeichen (COMP-3)
NCU - Rechenfeld ohne Vorzeichen (COMP)
NCS - Rechenfeld mit Vorzeichen (COMP)

NSU - Numerisch DISPLAY ohne Vorzeichen (DISP)
NSS - Numerisch DISPLAY mit Vorzeichen (DISP)
NSE - Numerisch druckaufbereitet

ABS - Alphabetisch (DISP)
ANS - Alphanumerisch (DISP)
ANSE - Alphanumerisch druckaufbereitet

GRP - Gruppenfeld
HEX - Standard-Ausgabe

TA TRIUMPH-ADLER
COB Ref 1282 d

Bei fehlerhafter Eingabe werden vom Debug-System folgende Meldungen ausgegeben:

	TA 1610 TA 1620	TA 1630
Falsche Stopadresse	5801 BREAKPNT?	-
Zuviele Stopadressen	5802 OVERFLOW!	-
Stopadresse schon definiert	5803 DUPLICATE!	-
Ungültiges Kommando	5804 COMMAND?	C?
Falsche Screen-Zeilen-Nummer	5805 LINE!	-
Falsche Programmadresse	5806 LOCATION!	-
Syntaxfehler	5807 SYNTAX!	S?
Daten abgeschnitten	5808 TRUNCATION	-
Falscher Datentyp	5809 TYPE?	-
Falscher Wert	5810 VALUE?	V?
Wartezustand	5811 WAIT	-

Fehlermeldungen des COBOL-Compilers

Meldungen	Warnung	Erläuterungen
ACCESS CLASH		Nichtsequentieller Zugriff wurde für eine sequentielle Datei angegeben.
ATTRIBUTE CLASH		Es widersprechen sich Angaben bezüglich Feldattributen in der Datenbeschreibung der Screen Section.
BLANK WHEN ZERO		BLANK WHEN ZERO wurde für ein nichtnumerisches oder Gruppenfeld angegeben.
CLASS		Der in der Klassenbedienung referenzierte Datentyp ist hier nicht erlaubt.
COPY		Die COPY-Anweisung kann wegen eines permanenten Fehlers bezüglich des gekennzeichneten Textnamens nicht ausgeführt werden.
COMPOSITE SIZE		Die in einer Anweisung verwendeten Operanden ergeben aufgrund Überlappung eine Länge größer als 18 Dezimalziffern.
DATA OVERFLOW		Der Datenbereich (Working-storage und Literale) ist länger als 65535 Bytes.
DATA TYPE		Der angegebene Datentyp ist bei dieser Referenz nicht erlaubt.
DECLARATIVE REFERENCE	W	Nur eine Warnung. Innerhalb des DECLARATIVE-Bereiches wird ein Prozedurname referenziert, der außerhalb des DECLARATIVES-Bereiches liegt.
DEVICE CLASH		Für ein Gerät ohne Direktzugriffsmöglichkeit wurde RANDOM definiert.
DEVICE TYPE		Die Referenzierung oder Beschreibung der Datei widerspricht dem Gerätetyp.
DOUBLE DECLARATION		Für eine Datei oder einen Bezeichner wurden **mehrfache** Erklärungen angegeben.
DOUBLE DEFINITION		Mehrere Definitionen für einen Bezeichner.
DUPLICATE	W	Nur eine Warnung. Mehrere USE-Prozeduren wurden für dieselbe Funktion angegeben.
EDIT PICTURE OVERFLOW		Editiertes Picture wird relativ zum Programmanfang "zu spät" referenziert (s. Kapitel 14.7).
FILE BOTTOM		Die Dateierklärung der angegebenen Datei enthält eine LINES AT BOTTOM Angabe mit einem Datennamen, der falsch gekennzeichnet oder nicht als ein numerisch ganzzahliges Datenfeld ohne Vorzeichen definiert ist.
FILE DESCRIPTION		Zu dem angegebenen Dateinamen existiert keine, oder eine falsche Dateierklärung (FD).

Fehlermeldungen des COBOL-Compilers

Meldungen	Warnung	Erläuterungen
FILE FOOTING		Die Dateierklärung der angegebenen Datei enthält eine WITH FOOTING-Angabe mit einem Datennamen, der falsch gekennzeichnet oder nicht als ein numerisch ganzzahliges Datenfeld ohne Vorzeichen definiert ist.
FILE LINAGE		Die Dateierklärung der angegebenen Datei enthält eine LINAGE IS-Angabe mit einem Datennamen, der falsch gekennzeichnet oder nicht als numerisch ganzzahliges Datenfeld ohne Vorzeichen definiert ist.
FILE NAME		Der angegebene Dateiname hat eine ungültige externe Dateinamenerklärung.
FILE NAME REQUIRED		In der I/O-Anweisung wird nicht, wie erforderlich, ein Dateiname referenziert.
FILE RECORD KEY		Der angegebene Dateiname besitzt einen RECORD KEY, welcher falsch qualifiziert ist oder welcher nicht als alphanumerisches Datenfeld innerhalb der Satzbeschreibung der betreffenden Datei definiert ist.
FILE RECORD SIZE		Der angegebene Dateiname besitzt eine Satzlänge, die mit der aktuellen Datenbeschreibung im Widerspruch steht.
FILE RELATIVE KEY		Der angegebene Dateiname besitzt einen RELATIVE KEY-Datennamen, der entweder falsch qualifiziert oder in der zugehörigen Satzbeschreibung definiert oder nicht als vorzeichenlose Ganzzahl beschrieben ist.
FILE SCREEN RECORD	W	Nur eine Warnung. Der angegebene Dateiname besitzt einen SCREEN IS-Datennamen, der entweder nicht definiert oder einer anderen Bildschirmdatei zugeordnet ist.
FILE STATUS		Der angegebene Dateiname besitzt ein Statusfeld, das entweder falsch qualifiziert oder nicht in der WORKING-STORAGE SECTION als zwei Zeichen langes, alphanumerisches Datenfeld definiert ist.
FILE TOP		Die Dateierklärung der angegebenen Datei enthält eine LINES AT TOP-Angabe mit einem Datennamen, der falsch gekennzeichnet oder nicht als ein numerisch ganzzahliges Datenfeld ohne Vorzeichen definiert ist.
FILE TYPE		Die Zugriffsart oder die Dateiorganisation widerspricht der gekennzeichneten Anweisung.
FILLER LEVEL		Es wurde ein nichtelementares FILLER Datenfeld definiert.

Fehlermeldungen des COBOL-Compilers

Meldungen	Warnung	Erläuterungen
GROUP CLASH		Die USAGE-, VALUE- oder FIELD ATTRIBUTE-Klausel der Gruppe widerspricht derselben Klausel für die Gruppe.
GROUP VALUE CLASH	W	Ein Datenfeld, das einer Gruppe angehört, welche mit VALUE IS beschrieben ist, ist mit der SYNCHRONIZED, JUSTIFIED oder USAGE (nicht USAGE IS DISPLAY) definiert.
IDENTIFIER		Ein Bezeichner ist nicht richtig angegeben oder der Bezeichner ist falsch oder doppelt definiert.
ILLEGAL PERFORM		Eine PERFORM-Anweisung referenziert einen undefinierten oder falsch qualifizierten Paragraphen, oder die Anweisung verletzt die Segmentierungsregeln.
INVALID ID		Es wird ein ungültiger Bezeichner referenziert.
INVALID PARAGRAPH		Es wird ein ungültiger Paragraphenname referenziert.
JUSTIFY		Die JUSTIFY-Klausel widerspricht anderen Attributen.
KEY REQUIRED		Für eine Relative-Datei im Direktzugriff (RANDOM) wurde kein RELATIVE KEY definiert, oder für eine Index-Datei wurde kein RECORD KEY definiert.
LABEL		Die angegebene oder fehlende LABEL RECORD-Klausel widerspricht den Geräte-Standards.
LEVEL	W	Die angegebene Stufennummer ist entweder in sich oder durch die Stellung in der Gruppe ungültig. Falls Stufennummer 01 oder 77 nicht im Bereich A beginnt, ist die Meldung nur eine Warnung.
LINAGE REQUIRED		Die FD-Erklärung der angesprochenen Datei enthält nicht die LINAGE-Klausel. Eine Angabe bei der WRITE-Anweisung setzt jedoch das Vorhandensein der Klausel voraus.
LINE-POSITION INVALID		Die LINE- und/oder POSITION-Angaben sind falsch. Entweder sind die Angaben außerhalb des logischen Bildschirmbereichs oder sie wurden für einen logischen Bildschirmbereich gemacht, der einer Scroll-Datei zugewiesen ist, oder die Angaben zeigen auf eine 'kleinere' Position als bei der Übersetzung des Bildschirmsatzes schon erreicht worden ist.

Fehlermeldungen des COBOL-Compilers

Meldung	Erläuterungen
LINKAGE	Ein Bezeichner in der USING-Klausel der PROCEDURE-Überschrift ist nicht in der LINKAGE SECTION enthalten oder eine Anweisung referenziert ein LINKAGE-Datenfeld, das keinem Bezeichner in der USING-Klausel der PROCEDURE-Überschrift untergeordnet ist.
LITERALE VALUE	Der Wert des Literals ist im angegebenen Kontext ungültig.
MOVE	Die Operanden einer MOVE-Anweisung definieren einen unzulässigen MOVE.
MUST BE INTEGER	Es muß eine Ganzzahl angegeben sein.
MUST BE PROCEDURE	Hier muß entweder als Referenz oder als Definition ein Prozedurname stehen, oder die Referenz muß sich auf einen Prozedurnamen beziehen, der nicht dem Declaratives-Bereich angehört.
MUST BE SECTION	Hier muß ein Kapitelname stehen.
NESTING	Bedingungen, die keine IF-Bedingungen sind, sind falsch geschachtelt.
NOT IN REDEFINE	Die VALUE IS-Klausel steht in einem REDEFINES-Datenfeld.
OCCURS	Die OCCURS-Klausel steht auf einem unzulässigen Level oder die Dimension der Tabelle ist größer drei.
OCCURS-VALUE CLASH	Die VALUE IS-Klausel und die OCCURS-Klausel sind für dasselbe Datenfeld definiert.
PICTURE	Die Syntax der PICTURE-Zeichenkette ist falsch.
PICTURE-BWZ CLASH	Bei der Datenbeschreibung ist BLANK WHEN ZERO angegeben und die PICTURE-Zeichenkette enthält das Stern-Zeichen.
PICTURE-USAGE CLASH	Die USAGE-Klausel widerspricht der angegebenen PICTURE-Zeichenkette.
PROCEDURE INDEPENDENCE	Eine PERFORM-Anweisung bezieht sich auf Prozeduren in unabhängigen Segmenten und nicht auf das aktuelle Segment.
PROGRAM OVERFLOW	Der Bereich der Anweisungen (PSECT) ist länger als 32 767 Bytes.

Fehlermeldungen des COBOL-Compilers

Meldungen	Warnung	Erläuterungen
RECORD KEY		Der RECORD KEY ist für eine Datei erklärt, die nicht mit ORGANIZATION IS SEQUENTIAL beschrieben ist, oder eine START-Anweisung mit der KEY-Angabe referenziert ein Datenfeld, welches nicht nach dem linken Byte des definierten Schlüssels ausgerichtet ist.
RECORD REQUIRED		In der WRITE- oder REWRITE-Anweisung muß ein Datensatz einer Datei referenziert werden.
REDEFINES		REDEFINES wurde innerhalb eines OCCURS-Bereiches angegeben oder es wird nicht das letzte Datenfeld redefiniert.
REDEFINES ERROR (SIZE)		Der referenzierte Datenname redefiniert ein Datenfeld das nicht dieselbe Anzahl von Zeichenpositionen besitzt und nicht auf Stufe 01 steht.
REFERENCE INVALID		Die angegebene Referenzierung ist hier nicht zulässig
RELATION		Die Operanden des Vergleichstests sind nicht vergleichbar.
RELATIVE KEY		RELATIVE KEY wurde für eine Datei definiert, die keine relative Dateiorganisation besitzt, oder eine START-Anweisung mit KEY-Angabe referenziert ein Datenfeld, das nicht mit dem definierten Schlüssel übereinstimmt.
RESERVED WORD CONFLICT	W	An einer Stelle, wo ein Anwenderwort erwartet wird, steht ein in COBOL reserviertes Wort oder Symbol. In der Fehlerstatistik gilt diese Meldung nur als Warnung, wenn ein in ANSI COBOL reserviertes Wort nicht in TA1600-COBOL als reserviert gilt.
SCAN RESUME	W	Nur eine Warnung. Der Prüfvorgang wurde durch den vorausgehenden Fehler beendet und an der unterstrichenen Position wieder fortgesetzt.
SECTION CLASH		Eine VALUE IS-Klausel steht in der FILE oder LINKAGE SECTION.
SEGMENT	W	Nur eine Warnung. Die Segmentnummer in einem unabhängigen Segment ist nicht dieselbe wie im aktuellen Segment oder die Nummer eines neuen unabhängigen Segmentes.
SEPARATOR	W	Nur eine Warnung. Redundante Interpunktion.

Fehlermeldungen des COBOL-Compilers

Meldungen	Warnung	Erläuterungen
SIGN		Die SIGN-Klausel widerspricht der USAGE- und PICTURE-Klausel.
SIZE	W	Nur eine Warnung. Die Größe eines referenzierten Datenfeldes ist in diesem Kontext falsch.
SIZE ERROR		Die erklärte Größe eines Satzes widerspricht der angegebenen Referenz.
SUBSCRIPT		Eine falsche Anzahl von Subskripten oder Indizes wurde verwendet.
SYNC		Die SYNCHRONIZED-Klausel wurde für ein Gruppenfeld verwendet.
SYNTAX		Ein falsches Zeichen oder ein falsches reserviertes Wort wurde verwendet.
UNDEFINED		Die Datei, die im FD-Paragraph referenziert wird, ist nicht definiert.
UNDEFINED PROCEDURE		Eine GO TO-Anweisung referenziert einen nicht definierten oder falsch qualifizierten Paragraphen.
USE REQUIRED		Nach einer Kapitelüberschrift im DECLARATIVES-Bereich steht keine USE-Anweisung. Die Fehlermeldung kann auch daraufhinweisen, daß im Programm die END DECLARATIVES-Anweisung vergessen wurde.
USING COUNT	W	Nur eine Warnung. Die Anzahl der Datenfelder in der USING-Liste einer CALL-Anweisung ist von der Anzahl der Datenfelder verschieden, die bei der ersten Bezugnahme auf denselben Programmnamen verwendet wurde.
VALUE SIZE		Der angegebene Wert in der VALUE IS-Klausel führt zum Abschneiden von Ziffern ungleich Null.
VALUE		Die VALUE IS-Klausel widerspricht anderen angegebenen Attributen.
VARIABLE RECORD	W	Nur eine Warnung. Eine READ... INTO-Anweisung referenziert eine Datei mit variabler Satzlänge.

TA 1620 COBOL-Systemfehlermeldungen

Meldung	Erläuterungen
COMPILER LOGIC FAILURE	Im Compiler ist ein interner Fehler erkannt worden. Das Problem ist dem Systemprogrammierer zu übergeben.
COMPILER MEMORY OVERFLOW	Zur Compilierung des Quellprogramms benötigt der Compiler mehr Speicher.
COMPILER POINTER OVERFLOW, nnnn	Die Compiler-Tabelle mit der Nummer nnnn ist übergelaufen.

TA 1620 COBOL-Systemfehlermeldungen

Fehler-code TA 1620	Meldungen (TA1630)	Erläuterungen	Fehlerbeseitigungen
	COBOL COMPILER TCA ERROR	Das System hat eine Fehlerbedingung in der TCA (Task Communication Area) erkannt.	Das Problem ist dem Systemprogrammierer zu übergeben.
	COBOL COMPILER TLF ERROR	Das System hat eine Fehlerbedingung beim Zugriff auf den TLF (Terminal Local File) erkannt.	Das Problem ist dem Systemprogrammierer zu übergeben.
7606	COBOL COMPILER BUG	Die Compiler konnte die Compilierung wegen einer internen Fehlerbedingung nicht beenden.	Das Problem ist dem Systemprogrammierer zu übergeben.
7600	COBOL COMPILER ABNORMAL TERMINATION	Der Compiler hat die Generierung des Objektmoduls aus dem Quellprogramm abgebrochen.	Aus der Druckliste ist die Ursache zu bestimmen.
7603	COBOL POINTER OVERFLOW	Eine der internen Zeigertabellen des Compilers ist übergelaufen.	Die X-Angabe ist zu unterdrücken (Crossreferenzliste); wenn der Fehler wieder auftritt: Programm kürzen!
7605	COBOL PROGRAM OVERFLOW	Zur Speicherung des Programms ist nicht genügend Speicher vorhanden.	Andere Programme sind zu beenden oder die X-Angabe ist zu unterdrücken (Crossreferenzliste).
	COBOL COMPILER OVERLAY LOAD ERROR	Beim Laden eines Compiler-Overlays ergab sich ein Fehler	Das Problem ist dem Systemprogrammierer zu übergeben.
	COPY FILE CLOSE ERROR	Bei dem Versuch, die Copydatei zu schließen, ergab sich ein Fehler.	Das Problem ist dem Systemprogrammierer zu übergeben.
	COPY FILE OPEN ERROR	Bei dem Versuch, die Copydatei zu eröffnen, ergab sich ein Fehler.	Das Problem ist dem Systemprogrammierer zu übergeben.

TA 1620 COBOL-Systemfehlermeldungen

Fehler-code TA 1620	Meldungen (TA1630)	Erläuterungen	Fehlerbeseitigungen
	COBOL COMPILER FATAL ERROR ...END VECTOR TAKEN	Das Betriebssystem hat einen schweren Fehler festgestellt und die Compilation beendet	Im BS-Programmierhandbuch können mögliche Task-Fehler nachgeschlagen werden
7607	COBOL SPILL (FILE) RECORD OVERFLOW	Die Plattenkapazität reicht für die auszulagernden Tabellen nicht aus.	Die Platte ist ggf. zu reorganisieren.
7601	FATAL SYNTAX ERROR OR NULL PROGRAM	Ein Syntaxfehler im Quellcode führte zu einem schweren Fehler im Compiler.	Prüfe die Quellanweisungen und korrigiere die Syntax. Wiederhole die Compilierung.
	INVALID COMPILE OPTION	Eine oder mehrere der ausgewählten Compilersteuerparameter für diesen Compilierungslauf sind nicht gültige Parameter.	Korrigiere den falschen Steuerparameter. (Der Compiler arbeitet auch ohne Korrektur richtig. Er ignoriert den ungültigen Parameter).
	TAXO I/O ERROR	Bei einem der EA-Geräte trat ein Fehler auf.	Man benutze Systemfehlercodes, um den Grund für den Fehler zu finden.
	LISTING FILE ERROR	Ein Fehler trat während der Übertragung der Daten zur Druckdatei auf.	Man benutze Systemfehlercodes, um den Grund für den Fehler zu finden.
	OBJECT FILE ERROR	Ein Fehler trat während der Übertragung der Daten zur Objektdatei auf.	Man benutze Systemfehlercodes, um den Grund für den Fehler zu finden.
7602, 7604	MEMORY REQUESTED EXCEEDS SYSTEM CAPACITY	Die Speicherkapazität des Systems ist nicht ausreichend, um die Speicheranforderungen zu befriedigen.	Andere Programme sind zu beenden oder die X-Angabe ist zu unterdrücken.
	ROLL SPILL FILE ERROR	Ein Fehler trat bei der Verarbeitung von Roll-Spill-Dateien auf.	Das Problem ist dem Systemprogrammierer zu übergeben.

TA TRIUMPH-ADLER
COB-Ref LRS.d

TA 1620 COBOL-Systemfehlermeldungen

Meldungen (TA1630)	Erläuterungen	Fehlerbeseitigungen
SOURCE FILE ERROR	Während der Übertragung der Daten von der Quelldatei trat ein Fehler auf.	Man benutze die Systemfehlercodes, um den Grund für den Fehler zu finden.
TOO MANY NESTED COPIES	Eine COPY-Anweisung wurde gelesen, die mehr als fünf geschachtelte Level enthält.	Das Quellprogramm ist so zu ändern, daß die Schachtelungstiefe der COPY-Anweisung weniger als fünf ist.

Fehlermeldungen des TA 1600 COBOL Laufzeitsystems

Fehlercode TA1620	Fehlermeldung (TA1630)	Erläuterungen	Fehlerbeseitigung
5000		Der vorher angezeigte E/A Fehler führte zum Abbruch des Programmes an dieser Stelle. Wurde vorher kein Fehler ausgegeben, dann lag die Ursache für den Fehler darin, daß • die ACCEPT/DISPLAY Datei in der Terminal-Partition nicht eröffnet werden konnte. • die ACCEPT/DISPLAY-Datei in der Non-Terminal-Partition eröffnet werden sollte • Laufzeitsystem und Compileroutput nicht verträglich	Überprüfen der Programmlogik bzw. der Partitionsgröße
5001	COMPILATION	Compilierungsfehler erkannt: Es wurde der Versuch gemacht eine fehlerhaft-compilierte Anweisung auszuführen.	Korrigieren der fehlerhaften COBOL-Anweisung und Recompilieren
5002		Fehler im aufgerufenen Programm: Der vorher angezeigte Fehler trat in dem von diesen Programm aufgerufenen Programm an der betreffenden Stelle auf.	Überprüfen des aufgerufenen (CALL) Programms

TA TRIUMPH-ADLER
COBOL Ref. Card

Fehlermeldungen des TA 1600 COBOL Laufzeitsystems

Fehlercode TA1620	Fehlermeldung (TA1630)	Erläuterungen	Fehlerbeseitigung
5012	INVALID ADDRESS	Fehlerhafte Adresse: Es wurde versucht eine ungültige Adresse zu referenzieren. Dies kann auftreten, wenn . beim Aufruf eines Unterprogramms weniger Argumente übergeben werden, als declariert bzw. referenziert werden oder wenn . FILE STATUS oder RELATIVE KEY als Linkage-Elemente während einer E/A Operation verwendet werden.	Überprüfen des Programmes, insbesondere hinsichtlich Programmübergabe bei CALL
5008	INVALID DOPE	Fehlerhafte Datendarstellung: Es wurde der Versuch gemacht eine Instruktion auszuführen, die ungültige Daten referenziert	Überprüfen der Compilierung, insbesondere bezgl. aufgetretener Fehler; Überprüfen der Objektdatei und gegebenfalls Recompilierung.
5009	INVALID EDIT	Fehlerhaftes Editierungspicture: Es wurde versucht eine Instruktion auszuführen, die eine ungültige Editierungsmaske referenziert.	Überprüfen der Compilierung, insbesondere bzgl. aufgetretener Fehler; Überprüfen der Objektdatei und gegebenenfalls Recompilierung.

Fehlermeldungen des TA 1600 COBOL Laufzeitsystems

Fehlercode TA1620	Fehlermeldung (TA1630)	Erläuterungen	Fehlerbeseitigung
5006	INVALID INSTRUCTION	Undefinierte Objektanweisung. Es wurde versucht eine ungültige Instruktion auszuführen.	Überprüfen der Compilierung, insbesondere bzgl. aufgetretener Fehler; Überprüfen der Objektdatei und gegebenfalls Recompilierung.
		oder Es wurde versucht, mit einem Programmiersystem ein Programm auszufüllen.	oder Laden eines Standardbetriebssystems
	external-name IO ERROR=ZZ.II AT:xxyyyy in pname	Fehler bei der Ausführung einer E/A-Anweisung: Bei der Ausführung einer E/A-Anweisung, für die keine AT END, INVALID KEY oder USE Klausel existierte oder anwendbar war, trat ein Fehler auf. Der zugehörige Dateiname ist "external-name", xxyyyy ist die Laufzeitadresse im Programm "pname".. ZZ ist der COBOL-File-Status und II der TAXO bzw. TASO Fehlercode.	Überprüfen des Programmes hinsichtlich Logik und Korrektheit der E/A; Korrigieren des Programmes und Recompilieren.
5005	LINAGE	Fehlerhafter LINAGE-Wert: Der bei WRITE ...ADVANCING angegebene Zeilenwert liegt außerhalb des LINAGE-Bereiches	Überprüfen des Programms und ggf. Recompilieren.
5004	LOAD OVERLAY (SVC) ERROR	Fehler beim Laden eines Overlays	Überprüfen der CALL-Schnittstellen bzw. Link-Kontroll-Datei (TA1630)

Fehlermeldungen des TA 1600 COBOL Laufzeitsystems

Fehlercode TAS0(TA1620)	Fehlermeldung (TA1630)	Erläuterungen	Fehlerbeseitigung
5003	SUBSCRIPT RANGE	Subscript-Bereichs-Fehler: Bei der Subskribierung wurde ein Wert angegeben, der nicht im Bereich der OCCURS-Angabe liegt, oder die Länge eines Operanden in einer Anweisung ist größer als 2047.	Überprüfen der Programmlogik, insbesondere bezgl. OCCURS und Subskribierung; ggf. Recompilieren.
5007	UNDEFINED LABEL REF	Fehlerhafter Bezug auf einen Paragraphennamen: Bei der Ausführung einer Anweisung wurde ein Paragraph referenziert, der nicht definiert ist. Dies kann auch dadurch hervorgerufen werden, daß der Paragraph in dem Sinne undefiniert ist, weil er im falschen Segment liegt - Regeln bzgl. Segmentierung verletzt.	Überprüfen des Programmes bzgl. Referenzen zwischen abhängigen und unabhängigen Segmenten; Recompilierung
5010		Fehler bei DTS-Benutzung: Das Laufzeitdebugsystem erzeugt einen Fehler während der Benutzung des Bildschirms für die Kommandoausgabe.	DTS-E/A im Programm und DTS-E/A des Laufzeitdebuggers konkurrieren miteinander; Abstimmen der E/A

Fehlermeldungen des TA 1600 COBOL Laufzeitsystems

Fehlercode TA1620	Fehlermeldung (TA1630)	Erläuterungen	Fehlerbeseitigung
5011		Fehlerhaftes Durchlaufen der USE-FORCHECK-Prozedur: Es wurde versucht die USE-FOR-CHECK-Prozedur erneut zu durchlaufen, obwohl diese noch nicht vollständig abgearbeitet war.	Überprüfen der Programmlogik bzgl. der Checkfeld-Verarbeitung
5013		USE-FOR-CHECK fehlt: Während der Verarbeitung eines READ-Befehls auf eine Bildschirmdatei wurde ein Checkfeld erkannt, eine zugehörige USE-FOR-CHECK Prozedur fehlt jedoch.	Korrigieren des Quellprogramms, Recompilieren
5014		Es wurde versucht, ein aktives Unterprogramm zu "canceln" oder ein aktives Unterprogramm mit CALL aufzurufen	Überprüfen der Programmlogik

TA 1630 COBOL-Laufzeit Systemfehlermeldungen

Meldungen	Erläuterungen	Fehlerbeseitigungen
BAD HEX VALUE ON OBJECT FILE	Die Objektdatei enthält ein Zeichen, welches keinen Hexadezimalwert darstellt.	Man überprüfe, ob die Objektdatei die richtige Datei ist und sie richtig positioniert ist. Falls notwendig, ist die Übersetzung zu wiederholen.
COBOL RUNTIME TCA ERROR	Das System stellte beim Arbeiten mit der TCA (task communication area) einen Fehler fest.	Das Problem ist dem Systemprogrammierer zu übergeben.
COBOL RUNTIME TLF ERROR	Das System stellte einen Fehler auf der TLF fest.	Der für den Parameter MESSAGE ACCESS NAME angegebene Wert ist unzulässig oder bezeichnet eine Datei, auf die-z.B. von einer anderen Station aus - schon zugegriffen wird
CHECKSUM ERROR	Eine Prüfsumme in der Objektdatei war falsch.	Man überprüfe, ob die Objektdatei die richtige Datei ist und sie richtig positioniert ist. Falls notwendig, ist die Übersetzung zu wiederholen.
END COBOL RUN	Das Programm wurde abgeschlossen.	
ERROR READING OBJECT FILE	Während des Lesevorganges für die Objektdatei wurde ein Fehler entdeckt.	Man bediene sich der Systemdienste.
INVALID TAG	Ein Kontrollzeichen in der Objektdatei ist ungültig.	Man überprüfe, ob die Objektdatei die richtige Datei ist und sie richtig positioniert ist. Falls notwendig, ist die Übersetzung zu wiederholen.
LOAD ADDRESS OUT OF RANGE	Die Objektdatei enthält eine Ladeadresse, die außerhalb der zulässigen Grenzen liegt.	Man überprüfe, ob die Objektdatei die richtige Datei ist und sie richtig positioniert ist. Falls notwendig, ist die Übersetzung zu wiederholen.

TA 1630 COBOL-Laufzeit Systemfehlermeldungen

Meldungen	Erläuterungen	Fehlerbeseitigungen
UNABLE TO GET MEMORY	Das System hat nicht genügend Speicher, um den Anforderungen des Programmes zu genügen.	Man beende andere Programme und wiederhole die Programmausführung.
UNABLE TO OPEN OBJECT	Das System stellt einen Fehler fest, als es die Objektdatei eröffnen wollte.	Das Problem ist dem Systemprogrammierer zu übergeben.
UNEXPECTED END OF FILE ON OBJECT FILE	Die Objektdatei enthält ein EOF-('end-of-file')-Kennzeichen vor den Daten, welche zur Ausführung des Programmes notwendig sind.	Man überprüfe, ob die Objektdatei die richtig Datei ist und sie richtig positioniert ist. Falls notwendig, ist di Übersetzung zu wiederholen.